Appendix

Schnorr's authentication scheme

Protocol

Schnorr's authentication scheme is based on the hardness of discrete log. Two prime numbers p and q are such that $q \mid p - 1$. A root of unity in Z_p of order q is a such that $\alpha \neq 1$, i.e., $\alpha^q \equiv 1\ (mod\ p)$. The prime numbers p, q and α are publicly known.

A secret key or identity consists of a private key s and a public key v. The private key s is a non-negative integer less than q, chosen uniformly at random. The public key is $v = \alpha^{-s}\ (mod\ p)$.

As with most zero knowledge authentication schemes, the core of the Schnorr protocol has the following main steps.

Initiation

The Prover initiates the protocol by sending the public key for which the prover claims to possess the corresponding private key.

Commitment by Prover.

The prover picks a non-negative number r less than q uniformly at random. The prover sends $x = \alpha^r\ (mod\ p)$ to the verifier.

Challenge from Verifier.

The verifier picks a non-negative number e less than 2^t, chosen uniformly at random. The number t is a security parameter that characterizes the soundness of the protocol. The verifier sends e to the prover.

Response from Prover.

The prover computes $y = r + s * e$ $(mod\ q)$. The prover sends y to the verifier. The verifier checks that $x = \alpha^y * v^e$ $(mod\ p)$ and accepts the prover's claim if and only if the equality is true.

Completeness.

It is clear that an honest verifier can engage in this protocol and prove its identity to the verifier with probability 1.

Soundness.

A fraudulent prover can cheat by guessing the correct challenge e ahead of time and sending the commitment $x = \alpha^r * v^e$ $(mod\ p)$ where the response is chosen to be $y = r$. However the probability of success for this attack is 2^{-t}. The success rate cannot be increased unless the discrete logarithm is easy to computed.

Zero knowledge.

This protocol is a proof of knowledge because two transcripts with the same commitment r can be generated, and is easy to compute the discrete logarithm ($log_{\alpha} v = (y - y')/(e - e')$ $(mod\ q)$). If there exists a fraudulent prover with non-negligible probability of success, then by repeatedly simulating this fraudulent prover it is possible to uncover two transcripts with the same commitment thus demonstrating that the protocol is a proof of knowledge.

Clearly, in the case of the honest verifier, it is possible to generate transcripts with the same distribution and hence the protocol is honest verifier zero knowledge. However, this protocol is not zero knowledge in the general case because a dishonest verifier could choose a challenge that is dependent on the commitment. This makes it difficult to generate transcripts with the same distribution. However, informally, the reason no information is revealed is that the numbers x and y are essentially random.